STATISTICAL PHYSICS FOR ADAPTIVE DISTRIBUTED CONTROL

David H. Wolpert

NASA Ames Research Center

http://ic.arc.nasa.gov/people/dhw/

NASA-ARC-03-097

THE GOLDEN RULE

DO NOT:

Find a value of a variable x,
that optimizes a function

INSTEAD:

Find a distribution over x,
that optimizes an expectation value

ADVANTAGES

1) Arbitrary data types.
2) Leverages continuous-space optimization.
("Gradient descent for symbolic variables".)
3) Akin to interior point methods.

- *Deep connections with statistical physics and game theory. So*
  - *Especially suited for distributed domains.*
  - *Especially suited for very large problems.*

ROADMAP

1) *What is distributed control, formally?*

2) *Review information theory*

3) *Optimal control policy for distributed agents*

4) *How to find that policy in a distributed way*

WHAT IS DISTRIBUTED CONTROL?

1) A set of N agents: Joint move $x = (x_1, x_2, \ldots, x_N)$

2) Since they are distributed, their joint probability is a product distribution:

$$\boxed{q(x) = \prod_i q_i(x_i)}$$

- *This definition of distributed agents is adopted from (extensive form) noncooperative game theory.*

EXAMPLE: KSAT

- $x = \{0, 1\}^N$
- A set of many disjunctions, "clauses", each involving K bits.
  E.g., $(x_2 \alpha x_6 \alpha \sim x_7)$ is a clause for K = 3
- Goal: Find a bit-string x that simultaneously satisfies all clauses. G(x) is #violated clauses.
- For us, this goal becomes: find a $q(x) = \prod_i q_i(x_i)$ tightly centered about such an x.

The canonical computationally difficult problem

ROADMAP

1) *What is distributed control, formally?*

2) *Review information theory*

3) *Optimal control policy for distributed agents*

4) *How to find that policy in a distributed way*

REVIEW OF INFORMATION THEORY

1) **Want a quantification of how "uncertain" you are that you will observe a value i generated from $P(i)$.**

2) **Require the uncertainty at seeing the IID pair (i, i') to equal the sum of the uncertainties for i and for i'**

3) **This forces the definition**

$$\boxed{\text{uncertainty}(i) = -\ln[P(i)]}$$

REVIEW OF INFORMATION THEORY - 2

4) So expected uncertainty is the *Shannon entropy*

$$\boxed{S(\mathrm{P}) \;\; \alpha \;\; -\alpha_i \, \mathrm{P}(i) \ln[\mathrm{P}(i)]}$$

- Concave over P
- α(P) is infinite at border of space of all P

5) *Information* in P, I(P), is what's left after the uncertainty is removed: -*S*(P).

ROADMAP

1) *What is distributed control, formally?*

2) *Review information theory*

3) *Optimal control policy for distributed agents*

4) *How to find that policy in a distributed way*

ITERATIVE DISTRIBUTED CONTROL

1) s is current uncertainty of what x to pick, i.e., uncertainty of where $q(x)$ is concentrated.
   - Early in the control process, high uncertainty.
2) Find q minimizing $E_q(G)$ while consistent with s.
3) Reduce s. Return to (2).
4) Terminate at a q with good (low) $E_q(G)$.

Can do (2)α (3) without ever explicitly specifying s

ITERATIVE DISTRIBUTED CONTROL - 2

1) The central step is to "find the q that has lowest $E_q(G)$ while consistent with $S(q) = s$".

2) So we must find the critical point of the Lagrangian

$$\boxed{L(q, T) = E_q(G) + T[s - S(q)]} \,,$$

i,e., find the q and T such that $\alpha L/\alpha q = \alpha L/\alpha T = 0$

- Deep connections with statistical physics (L is "free energy" in mean-field theory), economics

3) Then we reduce s; repeat (find next critical point).

EXAMPLE: KSAT

1) $S(q) = -\sum_i [b_i \ln(b_i) + (1 - b_i) \ln(1 - b_i)]$

where b_i is $q_i(x_i = \text{TRUE})$

2) $E_q(G) = \sum_{clauses\ j,\ x} q(x)\ K_j(x)$

$= \sum_{clauses\ j,\ x,\ i} \prod_i q_i(x_i)\ K_j(x)$

where $K_j(x) = 1$ iff x violates clause j

Our algorithm: i) Find q minimizing $E_q(G) - TS(q)$;

ii) Lower T and return to (i).

ROADMAP

1) *What is distributed control, formally?*

2) *Review information theory*

3) *Optimal control policy for distributed agents*

4) *How to find that policy in a distributed way*

DISTRIBUTED SEARCH FOR q

So control reduces to finding q such that $\alpha L/\alpha q = 0$

1) **Since the agents make their moves in a distributed way, that q is a product distribution.**

2) **But they must also find that q in a distributed way.**

3) **There are two cases to consider:**
   i) **Know functional form of G.**
   ii) **Don't know functional form of G - must sample.**

MINIMIZING L(q) VIA GRADIENT DESCENT

1) Each i works to minimize $L(q_i, q_{(i)})$ using only partial information of the other agents' distribution, $q_{(i)}$.

2) The $q_i(x_i)$ component of $\alpha L(q)$, projected onto the space of allowed $q_i(x_i)$, is

$$\alpha\, E_{q_{(i)}}(G \mid x_i) \;+\; ln(q_i(x_i))$$

$$—$$

$$\alpha dx q_i [\alpha\, E_{q_{(i)}}(G \mid x_i) \;+\; ln(q_i(x q_i))\,]$$

- The subtracted term ensures q stays normalized

GRADIENT DESCENT - 2

3) Each agent i knows its value of $ln(q_i(x_i))$.

4) Each agent i knows the $E_{q_{(i)}}(G \mid x_i)$ terms.

Each agent knows how it should change its q_i under gradient descent over $L(q)$

5) Gradient descent, even for categorical variables (!), and done in a distributed way.

6) Similarly the Hessian can readily be estimated (for Newton's method), etc.

EXAMPLE: KSAT

1) Evaluate $\boxed{E_{q_{(i)}}(G \mid x_i)}$ - the expected number of violated clauses if bit i is in state x_i - for every i, x_i

2) In gradient descent, decrease each $q_i(x_i)$ by

$$\boxed{\alpha[E_{q_{(i)}}(G \mid x_i) + T \ln[q_i(x_i)] - \text{const}_j]}$$

where α is the stepsize, and const_j is an easy-to-evaluate normalization constant.

3) We actually have a different T for each clause, and adaptively update all of them.

ADAPTIVE DISTRIBUTED CONTROL

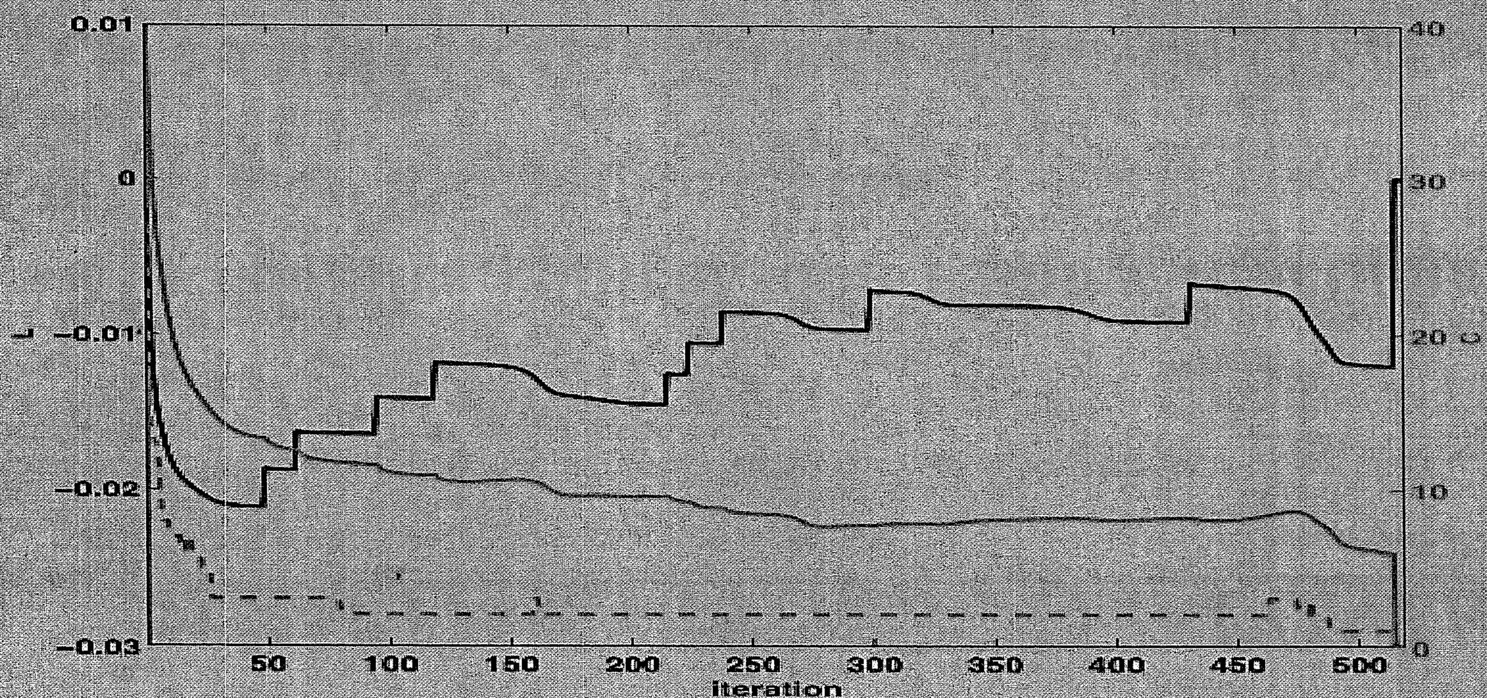
1) In *adaptive* control, don't know functional form of G(x). So use Monte Carlo:

- Sample G(x) repeatedly according to q;
- Each i independently estimates $E_{q_{(i)}}(G \mid x_i)$ for all its moves x_i;
- Only 1 MC process, no matter how many agents

So each q_i can adaptively estimate its update

EXAMPLE: KSAT

i) Top plot is Lagrangian value vs. iteration;
ii) Middle plot is average (under q) number of constraint violations;
iii) Bottom plot is mode (under q) number of constraint violations.



CONCLUSION

1) *A distributed system is governed by a product distribution q, by definition.*

2) *So distributed adaptive control is adaptive search for the q that optimizes $E_q(G)$.*

3) *That search can be done many ways, e.g., gradient descent, with or without Monte Carlo sampling.*